```
% Example of how to use LIBSVM on 64 bit Matlab installations.
%% Start script
clear all;close all;  % Clear all variables from memory and close all figures

X = [-1 -1; % XOR Data
     -1 +1;
     +1 -1;
     +1 +1];
Y = [-1;
     +1;
     +1;
     -1]; % Targets
N = length(Y); % Number of samples

%% Normally data is split into 5 folds (this has not been done here!)
% Define training data
test_range = 1:N;
train_range = [1:N];
train_data = X(train_range,:);
train_class = Y(train_range);
%Define testing data
test_data = X(test_range,:) ;
test_class = Y(test_range);

%%  Define SVM Kernel Hyperparameters
C = 10;  % Define C (box constraint)
G = 2;    % Define Gamma = 1/2*Sigma.^2 (width of Rbf)
%% "Grid search" ideally needed but not included here; refer to next page for
%% train_params
train_params = ['-s 0 -t 2 -g ' num2str(G) ' -r 0 -c ' num2str(C)];

%% Train SVM
model = svmtrain(train_class, train_data , train_params);

%% Test output (cross-validation not used here - must add!)
[predicted_label, accuracy, decision_values]= svmpredict(test_class,
test_data, model);

%% Plot results
figure(1);clf;hold;
for n=1:N,
    if predicted_label(n)==1,
plot(X(n,1),X(n,2),'bd','markersize',8,'linewidth',2,'markerfacecolor','b');
    else
plot(X(n,1),X(n,2),'ro','markersize',8,'linewidth',2,'markerfacecolor','r');
    end
end
axis('square');grid
axis([-2 2 -2 2]);
line([-2 2],[0 0],'LineWidth',1.5,'Color',[0 0 0]);
line([0 0],[-2 2],'LineWidth',1.5,'Color',[0 0 0]);
xlabel('x_1','fontsize',16);
ylabel('x_2','fontsize',16);

%% Only points plotted here, best to include the optimal separating
%% hyperplane and canonical planes if you can!!!
```

```
%%%%%%%%%%%%%%%% LIBSVM options %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% options:
% -s svm_type : set type of SVM (default 0)
%   0 -- C-SVC
%   1 -- nu-SVC
%   2 -- one-class SVM
%   3 -- epsilon-SVR
%   4 -- nu-SVR
% -t kernel_type : set type of kernel function (default 2)
%   0 -- linear: u'*v
%   1 -- polynomial: (gamma*u'*v + coef0)^degree
%   2 -- radial basis function: exp(-gamma*|u-v|^2)
%   3 -- sigmoid: tanh(gamma*u'*v + coef0)
% -d degree : set degree in kernel function (default 3)
% -g gamma : set gamma in kernel function (default 1/num_features)
% -r coef0 : set coef0 in kernel function (default 0)
% -c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
% -n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (def 0.5)
% -p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
% -m cachesize : set cache memory size in MB (default 100)
% -e epsilon : set tolerance of termination criterion (default 0.001)
% -h shrinking: whether to use the shrinking heuristics, 0 or 1 (default 1)
% -b probability_estimates: whether to train a SVC or SVR model for
probability estimates, 0 or 1 (default 0)
% -wi weight: set the parameter C of class i to weight*C, for C-SVC (def 1)
% model = svmtrain(training_label_vector, training_instance_matrix [,
'libsvm_options']);
%         -training_label_vector:
%             An m by 1 vector of training labels (type must be double).
%         -training_instance_matrix:
%             An m by n matrix of m training instances with n features.
%             It can be dense or sparse (type must be double).
%         -libsvm_options:
%             A string of training options in the same format as that of
LIBSVM.
%
% matlab> [predicted_label, accuracy, decision_values/prob_estimates] =
svmpredict(testing_label_vector, testing_instance_matrix, model [,
'libsvm_options']);
%
%         -testing_label_vector:
%             An m by 1 vector of prediction labels. If labels of test
%             data are unknown, simply use any random values. (type must be
double)
%         -testing_instance_matrix:
%             An m by n matrix of m testing instances with n features.
%             It can be dense or sparse. (type must be double)
%         -model:
%             The output of svmtrain.
%         -libsvm_options:
%             A string of testing options in the same format as that of
LIBSVM.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```